Section: *(circle one)* 01 (Vassileva) / 03 (Olorunleke) / 05 (Koehn)

Name: _

Student

Signatur

University of Saskatchewan
Department of Computer Science

CMPT 116 Midterm Examination

November 10, 2004

Total Marks: 54 marks **CLOSED BOOK** **Time: 60 Minutes**

PLEASE ANSWER QUESTIONS 1 through 17 ON THE PROVIDED OPTIC SCAN SHEET. Darken with pencil (HB preferred) the circle that corresponds with your answer for each of these questions. Be sure to include your name and student number on the top of the OPTIC SCAN sheet (and on the top of this question sheet).

THE REMAINDER OF THE QUESTIONS, QUESTIONS 18 through 20, ARE TO BE ANSWERED ON THIS QUESTION SHEET.
If you don't have enough space, write on the back of the page, indicating clearly that your answer is continued there. Be sure to pace yourself throughout the exam.

I. True / False Questions: *(1 mark each)* TO BE ANSWERED ON THE OPTIC SCAN SHEET.

ANSWERS: (A) TRUE (B) FALSE

✓ 1. In C++, a condition (logical expression) is considered **true** if it evaluates to any number, positive or negative, other than zero. If an expression evaluates to zero, it is **false**.

X 2. The following condition will always evaluate to **true** for integer values

```
(A < B) || (B > A)
 4 < 3      3 > 4
   0    or    0  ⇒ 0
```

✓ 3. If a C++ program compiles successfully, it may still contain errors.

✓ 4. In C++, it is possible to write every type of loop as a **while** loop or as a **for** loop.

X 5. The following statements interchange the contents of variables A and B:

```
A = B;   B = A;
```

X 6. If the initializing statement is missing in a **for** statement, a compiler error will result.

X 7. The following function returns the cube, x^3, of an integer value

```
void cube (int x) {
   int result = x * x * x;
   cout << result;  }
```

II. Multiple Choice Questions: *(2 marks each)* TO BE ANSWERED ON THE OPTIC SCAN SHEET.

*** SELECT THE SINGLE BEST CHOICE FOR EACH OF THE FOLLOWING QUESTIONS ***

8. Which of the following assignments will make the following expression true?

```
((a || !c) && (!b || c))
   T  T       T    T  T
```

I. a=1, c=1, b=any value ✓
II. b=0, c=0, a=any value ✓
III. a=1, b=0, c=any value ✓

[✓ (A) I, II, and III]
(B) I and II only
(C) I only
(D) II only
(E) III only

✓

9. After the statements

```
int team_reward=39, team_size=4;
double ind_reward=(double)(team_reward/team_size);
                    9.0   ←        39        4
```

have executed, what is the value of **ind_reward**?

(A) 9
[✓ (B) 9.0]
(C) 9.75
(D) 10
(E) 10.0

✓

10. Consider the following expression:

```
a + 5 * a - 8
```

Which of the following expressions with parentheses shows the order in which C++ will evaluate the above expression?

(A) `(a+5)*(a-8)`
(B) `((a+5)*a)-8`
(C) `(a)(+5)*(a)(-8)`
[✓ (D) `(a+(5*a))-8`]
(E) `((a)+(5))*((a)-(8))`

✓

11. An **argument** is:

(A) a variable that has been declared but not initialized
(B) a variable without a data type
[✓ (C) the data passed to a function in a function call]
(D) the data required by a function that is listed in the function header
(E) a variable that is initialized with a value and cannot be changed

✓

12. What does the function **F** do when called with the second argument, **n**, being greater than or equal to zero?

```
double F(double x, int n)
{
   if (n==0) return 0;
   return (x + x + F(x, n-1));
}
```

2x + 2x + 2x . . . (n times

(A) It returns: n^{2x}
(B) It returns: x^{2n}
(C) It returns: the sum of squares, x through n
[✓ (D)] It returns: 2nx
(E) It returns: n!

✓

13. What value will be returned by the function call **`mystery(10, 30)`**?

```
int mystery( int n1, int n2 ) {
  int i, y=1;

  for (i=2; i<= n1; i++)
    if (((n1 % i) == 0) && ((n2 % i) == 0))
      y = i;

  return y;
}
```

10 30 10

i	n1=10	n2=30	y=
2			2
3			2
.			
.			
10	10%10==0 ✓	30%10==0 =>	y = 10

✓

(A) 0
(B) 1
(C) 3
(D) 5
[✓ (E)] 10

14. A do-while statement

(A) automatically performs a validity check on data
(B) will always execute a fixed number of times
(C) may never execute
[✓ (D)] will always execute at least one time
(E) is the most efficient type of loop in C++

✓

15. The statement that forces an immediate exit from a switch, while, for, or do-while statement is:

(A) `termination`
[✓ (B)] `break`
(C) `quit`
(D) `continue`
(E) `function call`

✓

16. What is the output of the following fragment of code:

```
int sum = 0, int i;
for( i=0; i<5; i=i+2 )
      sum += i;
cout << i << ", " << sum << endl;
```

i	Sum (end of loop)
0	0
2	2
4	6
6 — doesn't run	

[✓ (A)] 4, 6
(B) 6, 12
(C) 4, 12
[✓ (D)] 6, 6
(E) none of the above

✗

17. When is the following code going to raise a flag for alarm (set alarm to true):

```
if (load/stress > 1.0)
   if (brittle)
       alarm = danger = true;
   else {
       alarm = false;
       danger = true; }

else
   if (brittle) {
       alarm = false;
       danger = true;}
   else
       alarm = danger = false;
```

✓

(A) When the material is brittle (brittle > 0)
(B) When the load exceeds the stress
[✓ (C)] When the load exceeds the stress and the material is brittle
(D) When the load doesn't exceed the stress and the material is brittle
(E) Always

6.5 /7

III. Analyzing & Writing Code: TO BE ANSWERED ON THIS QUESTION SHEET.

Question 18 *(7 marks)*

Trace the following code segments and indicate what the output is for each segment.

a)
```
int x, y;
for (x=0; x<4; x++)
{  for (y=x; y>0; y--)
        cout << x*y << ", ";
   cout << endl; }
```

2.5

Output:
```
1,          (½)
4, 2,       (½)
9, 6, 3,    (½)
```

b)
```
int Fun1(int a, int &b, int &c);
int Fun2(int a, int b);

void main ()
{  int a=1, b=2, c=3;
   a = Fun1(a, b, c);    a=12   b = 16   c = 4
   cout << a << ", " << b << ", " << c << endl;
} // end fn. main

int Fun1(int a, int &b, int &c)
{  c = a + c;                       c =4
   a = Fun2(a, b);         a=8
   b = a + 2 * c;          b = 8+2*4 =16
   cout << a << ", " << b << ", " << c << endl;
   return b - a + c;   16-8+4 = 12
} // end fn. Fun1

int Fun2(int a, int b)
             1      2
{  int c;
   b = a + b;     b=1+2 =3
   c = (b * b) - (a * a);    c= 9-1 =8
   cout << a << ", " << b << ", " << c << endl;
   return c;
} // end fn. Fun2
```

4

Output:
```
1, 3, 8
8, 16, 4
12, 16, 4
```

10/10

Question 19 *(10 marks)*

The head of the engineering labs has instructed you to write a program that can be used to make decisions about which company to buy some lab equipment from. Just under two hundred companies have submitted bids which have been collated into a text file where **the first column** contains **the company name** (assume company names do not have spaces between them e.g. *HomeDepot* rather than *Home Depot*), **the second column** contains **the cost** of the equipment, and **the third column** contains **a discount** which the company is willing to give. A portion of this file is shown below ("bids.txt"):

```
HomeHardware        25000.00     2500.00
CanadianTire         24650.99     0.00
    :
```

Now write a C++ program that prints the company name with the lowest final amount (after the discount has been applied). If there is more than one company with an equal lowest amount, then printing any one of these companies is good enough.

```
#include <iostream>
# include <fstream>
# include <io manip>


int   main() {
  ifstream  inPut;
  inPut.open("bids.txt");
  if (inPut.fail()) {
      cout << "error opening file. Ensure file is in correct directory.";
      return -5;
  }
  float lowestAmount = 1000000000000.0;
  char  comp[64];
  while(!(inPut.eof())) {
      Char tmp Name[64];
      flot bid, discount;
      inPut >> tmpName >> bid >> discount;
      if((bid-discount) < lowestAmount) {
            lowestAmount = bid -discount;
            comp = tmpName;
      }
  }
  inPut.close();
  cout << "The lowest bidding company is: " << comp << endl;
  return 0;
}
```

10/10

Question 20 *(10 marks)*

In this question, you are to construct your own function definition for $\mathbf{x}^{\mathbf{y}}$. For example, $2^3 = 8$, $3^2 = 9$, $2^{-2} = 0.25$ Note that you are **not** permitted to use the built-in function pow(a, b).

a) Construct an ITERATIVE version for your function definition of $\mathbf{x}^{\mathbf{y}}$

```
float power (intx, int y) {


    if(y>=0) {
        int result=1;
        for (int i =0 ; i<y ; i++)
            result *= x ;
        return result; }
    else{
        int invres =1;
        for (int i=0; i<y; i++)
            inv res*=x;
        return 1.0/invres; }
}
```

b) Now, construct a RECURSIVE version for your function definition of $\mathbf{x}^{\mathbf{y}}$

```
float power(float x, int y) {
    if(y== 0)
        return 1;
    else if(y>0)
        return x * power (x, y-1);
    else
        return (1.0/x) * power (x, y+1);
}
```

c) Provide a code segment within the body of main to demonstrate a (potentially useful) function call to one of the functions you constructed in part a) or b).

```
int main () {
  cout<< "enter x & y";
  int x, y;
  cin >> x >> y;
  cout << "x to the power y = "<< power(x, y);
  return 0;

} // end fn. main
```

/* ~ ~~~~~~~~~ ~ *The End* ~ ~~~~~~~~~ ~ */